Bayesian Approach for Software Reliability Growth Model with Random Cost

Hee Soo Kim
School of Aerospace and Naval Architecture, Chosun University
Gwangju 501-759, Korea

Mi Young Shin
Department of Mathematics, Catholic University of Korea
Puchon 420-743, Korea

Dong Ho Park
Department of Information and Statistics, Hallym University
Chuncheon 200-702, Korea

Abstract

In this paper, we generalize the software reliability growth model by assuming that the testing cost and maintenance cost are random and adopts the Bayesian approach to determine the optimal software release time. Numerical examples are provided to illustrate the Bayesian method for certain parametric models.

1. Introduction

As the computer system becomes more complex and multiple-function oriented, the developers of software systems are required to produce more reliable software products to prevent the computer systems from stopping its operations during the mission period.

Since most of the computer system failures are very often caused by the software system failures, rather than by the hardware systems, it is of great importance to apply the best possible software development policy to produce the most reliable software system.

To achieve such a goal, it is a general practice for the software developers to carry out the preliminary testing during a certain length of time before the software system is released for the operational use. During such a testing phase, an effort is given to detect and debug the faults latent in the software system so that the possibility of software failures can be reduced during the operational phase.

Although the number of faults detected and debugged during the testing phase is proportionally increased as the length of software testing becomes longer, other factors including the testing cost, delivery schedule and the competition with other developers, must be taken into account to determine the optimal length of testing phase as well.

Therefore it is very important that we decide the optimal length of software testing, which is called optimal release time. These optimal software release problems have been studied by many researchers [3-4,8-10]. For example, Pham [9] develops a cost model with an imperfect debugging and random life cycle as well as a penalty cost that is used to determine the optimal release policies for a software system and Kimura et al. [3] discuss several optimal release problems by using the maintenance cost model and consider the concept of present value into the cost factors and warranty period in the operational phase

In this paper, we extend the Kimura et al. [3] optimal software release problem by assuming the testing cost and maintenance cost are random.

Although most of the optimal software release problems have been discussed within the non-Bayesian framework, there exist a few Bayesian discussions in the literature [5-7].

In Section 2, we describe the software reliability growth model based on a non-homogeneous Poisson Process. The formula to calculate the total expected software cost is derived and the optimal software release time is determined in Section 3. Numerical examples are illustrated in Section 4.

2. Software reliability growth model

It is quite desirable that when the debugging activity is performed to fix the software faults detected, the faults are perfectly eliminated from the software system and no faults are newly introduced into the system and thereby the software reliability improves. In this paper, we discuss a software reliability model based on an NHPP (Non-homogeneous Poisson Process).

Let $N(t)$ be the cumulative number of faults detected up to time t and let $h(t)$ denote the intensity function of an NHPP which describes the software fault detection phenomenon as follows.

$$\Pr\{N(t)=n\} = \frac{e^{-m(t)}\{m(t)\}^n}{n!}, \quad \mathrm{n} = 0,1,\cdots,$$

$$m(t) = E[N(t)] = \int_0 h(\tau)d\tau,$$

where $m(t)$ denotes the expected number of faults detected during $(0,t]$ which is called a mean value function.

As for the intensity function of NHPP which represents the fault detection rate per unit time, we apply an exponential software reliability growth model [1] based on the following intensity function and discuss the optimal software release time

$$h(t) = \alpha\beta e^{-\beta t}, \quad \alpha > 0, \quad \beta > 0, \qquad (1)$$

where the parameters α and β are the expected number of initial faults latent in the software and the fault detection rate per fault, respectively.

We investigate the optimal release policy under the software reliability growth model (SRGM) with intensity function $h(t)$ of (1).

3. Bayesian method for SRGM

The following notations are adopted in this paper.

C_0	initial testing cost which is the barest minimum requirement
C_t	testing cost per unit time
C_w	maintenance cost per fault during the warranty period
T	software release time
$T*$	optimal software release time
T_w	warranty period
γ	discount rate of the cost

3.1 The cost model

We discuss optimal software release problems which consider both a testing cost C_t and maintenance cost C_w that the developer has to pay to fix any faults detected during the warranty period. For the non-Bayesian solution for the optimal software release time, the testing and maintenance cost are determined to exactly specific amount. However, it is quite likely that these costs are unknown in most of practical situations. Actually, it is more realistic that we assume the restrictive information such as the upper and the lower level of these costs or the tendency of spending of these costs is given.

Therefore, we discuss the Bayesian optimal software release policy in this paper. That is we assume testing and maintenance cost are random and the initial cost is assumed to be a known constant.

Then, the total expected software cost C(T) can be presented as:

$$C(T) = C_0 + E(C_t)\int_0^T e^{-\gamma t}dt + E(C_w(T)) \quad (2)$$

where $C_w(T)$ is the maintenance cost during the warranty period.

We can get the optimal software release time T^* by minimizing expected software cost in equation (2).

3.2 Bayesian Optimal software release policies

In this paper we assume that the length of the warranty period is constant and discuss the following two cases in term of the behavior of the maintenance cost $C_w(T)$ during the warranty period.

[Case 1] When the software reliability growth is not assumed to occur after the testing phase, $E(C_w(T))$ is given by

$$E(C_w(T)) = E(C_w)\int_T^{T+T_w} h(T)e^{-\gamma t}dt \quad (3)$$

Substituting (3) into (2), we have

$$C_1(T) = C_0 + E(C_t)\int_0^T e^{-\gamma t}dt + E(C_w)h(T)\int_T^{T+T_w} e^{-\gamma t}dt \quad (4)$$

Differentiating the equation (4) with respect to T and set it equal to 0, we have the equation

$$h(T) = \frac{E(C_t)}{E(C_w)((\beta+\gamma)/\gamma)(1-e^{-\gamma T_w})} \quad (5)$$

where $h(t) = \alpha\beta e^{-\beta t}$ is given as in (1). Solving the equation (5) for T, we obtain the solution T_1 as follows.

$$T_1 = \frac{1}{\beta}\ln\left[\frac{\alpha\beta E(C_w)((\beta+\gamma)/\gamma)\left(1-e^{-\gamma T_w}\right)}{E(C_t)}\right] \quad (6)$$

Since $h(T)$ is a decreasing function of T for $T \geq 0$, there exists a finite unique solution T_1 given in (6) if

$$h(0) > E(C_t)/\{E(C_w)((\beta+\gamma)/\gamma)(1-e^{-\gamma T_w})\}.$$

Since $C_1(T)$ is a convex function with respect to T, it achieves the minimum at $T^* = T_1$. If

$$h(0) \leq E(C_t)/\{E(C_w)((\beta+\gamma)/\gamma)(1-e^{-\gamma T_w})\},$$

then $C_1(T)$ is a monotonically increasing function of T. It is clear that $C_1(T)$ is minimized at $T = 0$ and thus $T^* = 0$ in this case.

Therefore, when the length of the warranty period is constant and the software reliability growth is not assumed to occur after the testing phase, the optimal software release policy is obtained as follows. If $h(0) > E(C_t)/\{E(C_w)((\beta+\gamma)/\gamma)(1-e^{-\gamma T_w})\}$, the optimal release time is $T^* = T_1$. Otherwise, $T^* = 0$

[Case 2] When the software reliability growth is assumed to occur after the testing phase, $E(C_w(T))$ in (3) is given by

$$E(C_w(T)) = E(C_w)\int_T^{T+T_w} h(t)e^{-\gamma t}dt, \quad (7)$$

and the total expected software cost is rewritten as

$$C_2(T) = C_0 + E(C_t)\int_0^T e^{-\gamma t}dt + E(C_w)\int_T^{T+T_w} h(t)e^{-\gamma t}dt \quad (8)$$

Differentiating the equation (8) with respect to T and set it equal to 0, we have the equation

$$h(T) = \frac{E(C_t)}{E(C_w)\left(1-e^{-(\beta+\gamma)T_w}\right)}, \quad (9)$$

Solving the equation (9) for T, we obtain the solution T_2 as follows.

$$T_2 = \frac{1}{\beta}\ln\left[\frac{\alpha\beta E(C_w)\left(1-e^{-(\beta+\gamma)T_w}\right)}{E(C_t)}\right]. \quad (10)$$

By using the same procedure as Case 1, $C_2(T)$ achieves the minimum at $T^* = T_2$ if

$$h(0) > E(C_t)/\{E(C_w)(1-e^{-(\beta+\gamma)T_w})\}.$$

When $h(0) \le E(C_t)/\{E(C_w)(1-e^{-(\beta+\gamma)T_w})\}$, $C_2(T)$ is a monotonically increasing function of T.

Therefore, when the length of the warranty period is constant and the software reliability growth is assumed to occur after the testing phase, the optimal software release policy is obtained as follows.

If $h(0) > E(C_t)/\{E(C_w)(1-e^{-(\beta+\gamma)T_w})\}$, the optimal release time is $T^* = T_2$. Otherwise, the optimal release time is $T^* = 0$.

3.3 Prior distribution

First, as a prior distribution for the testing cost per unit time $C_2(T)$, we assign a truncated normal distribution, $TN(\mu_c, \sigma_c^2, c, d)$. The prior distribution and the expectation for C_t are given by

$$f_{C_t}(c_t) = \frac{\frac{1}{\sqrt{2\pi}\sigma_c} e^{\{-(c_t-\mu_c)^2/2\sigma_c^2\}}}{\Phi\left(\frac{d-\mu_c}{\sigma_c}\right) - \Phi\left(\frac{c-\mu_c}{\sigma_c}\right)} I_{[c,d]}(c_t),$$

$$E[C_t] = \mu_c + \left[\frac{\phi\left(\frac{c-\mu_c}{\sigma_c}\right) - \phi\left(\frac{d-\mu_c}{\sigma_c}\right)}{\Phi\left(\frac{d-\mu_c}{\sigma_c}\right) - \phi\left(\frac{c-\mu_c}{\sigma_c}\right)}\right]\sigma_c,$$

where $\mu_c > 0$, $\sigma_c > 0$, $c \le c_t \le d$, ϕ is the standard normal pdf and Φ is the standard normal cdf.

For the prior distribution for C_w, which is maintenance cost per fault during the warranty period, we define a discrete distribution by using a discrete beta density on (c_L, c_U). It is well known that the discrete beta distribution allows for great flexibility in representing prior uncertainty [2] and so we assume the discretization of the beta density on (c_L, c_U) as a prior for C_w. The beta density defined as

$$g(u) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{(u-c_L)^{a-1}(c_U-u)^{b-1}}{(c_U-c_L)^{a+b-1}}, \quad 0 \le c_L \le u \le c_U$$

where $c_L, c_U, a, b > 0$.

The prior distribution for C_w is given by

$$P_j = \Pr(C_w = c_j) = \int_{c_j-\delta/2}^{c_j+\delta/2} g(u)\,du,$$

where $c_j = c_L + \delta(2j-1)/2$ and $\delta = (c_U - c_L)/m$ for $j = 1, 2, \cdots, m$.

4. Numerical Examples

In this section, we present several numerical examples to illustrate the optimal software release policies based on the Bayesian approach, which is derived in Sections 3. We investigate the sensitivity of testing cost and warranty period affecting the optimal release time and its corresponding expected software maintenance cost.

We assume the parameters as follows and consider several values of the warranty period, $T_w = 5, 10, 20, 50, 100$ with $m = 20$.

$\alpha = 1000$, $\beta = 0.05$, $c_0 = 1000$, $\gamma = 0.001$

The numerical results are summarized in Table 1 and Table 2 where the first entry shows the optimal release time T^* and second entry shows the corresponding total expected cost. Figure 1 illustrates these results in Case 2 when $T_w = 5$, $C_t \sim TN(300, 100^2, 250, 500)$ and $C_w \sim$ *discrete* Beta(a,b). Considering this situation with $C_w \sim$ *discrete* Beta(2,2), we have the optimal release time T^*=20.6592.

From the Table 1 and 2, we can see the following facts.

1) The impact of the warranty time T_w:

The optimal release time and total expected cost also increase when the warranty period gets longer. That is, we need to spend more time to test the software in order to have a longer warranty period.

2) The impact of the software reliability growth after the testing phase warranty time:

Comparing Table 1 and 2, it can be seen that the optimal release time of Case 2 is always shorter than that of Case 1. That is, the optimal release time is shorter when the software reliability growth is assumed to occur after the testing phase.

3) The impact of the prior distribution of the testing cost per unit time:

Increasing the mean value of the testing cost C_t will result in a shorter optimal release time.

Table 1. Optimal release time (the first entry) and the total expected software cost (the second entry) with $C_t \sim TN(\mu_t, \sigma_t^2, 250, 500)$, $C_w \sim$ *discrete* Beta(a,b) when $50 \le c_j \le 100$ in Case 1

T_w	C_t: $(\mu_t,\sigma_t^2)=(300,50^2)$		$(\mu_t,\sigma_t^2)=(300,100^2)$		$(\mu_t,\sigma_t^2)=(400,50^2)$	
			$(a=2,\ b=2)$			
5	23.1156	13615.16	23.1051	13618.78	17.3837	15599.49
10	36.9286	17561.46	36.9181	17567.20	31.1967	20885.74
20	50.6918	21439.69	50.6813	21447.51	44.9599	26080.79
50	68.7194	26439.46	68.7088	26449.97	62.9874	32778.21
100	82.0886	30089.50	82.0780	30101.98	76.3566	37667.60
			$(a=2,\ b=3)$			
5	21.7358	13217.94	21.7252	13221.35	16.0038	15067.40
10	35.5488	17169.69	35.5382	17175.22	29.8168	20360.95
20	49.3120	21053.27	49.3014	21060.89	43.5800	25563.17
50	67.3395	26059.95	67.3290	26070.26	61.6076	32269.84
100	80.7087	29715.03	80.6982	29727.30	74.9768	37165.98
			$(a=3,\ b=2)$			
5	24.4064	13986.24	24.3958	13990.05	18.6744	16096.57
10	38.2194	17927.45	38.2088	17933.38	32.4874	21375.99
20	51.9826	21800.67	51.9720	21808.69	46.2506	26564.35
50	70.0101	26793.99	69.9996	26804.70	64.2782	33253.12
100	83.3793	30439.33	83.3688	30451.99	77.6474	38136.21

Table 2. Optimal release time (the first entry) and the total expected software cost (the second entry) with $C_t \sim TN(\mu_t, \sigma_t^2, 250, 500)$, $C_w \sim$ *discrete* Beta(a,b) when $50 \le c_j \le 100$ in Case 2

T_w	C_t: $(\mu_t,\sigma_t^2)=(300,50^2)$		$(\mu_t,\sigma_t^2)=(300,100^2)$		$(\mu_t,\sigma_t^2)=(400,50^2)$	
			$(a=2,\ b=2)$			
5	20.6697	12910.69	20.6592	12913.93	14.9378	14655.83
10	32.1448	16200.92	32.1343	16205.93	26.4129	19063.23
20	41.5511	18869.95	41.5405	18876.39	35.8191	22638.51
50	48.8694	20929.24	48.8589	20936.79	43.1375	25397.02
100	50.3731	21350.49	50.3626	21358.26	44.6412	25961.30
			$(a=2,\ b=3)$			
5	19.2899	12512.50	19.2794	12515.53	13.5579	14122.43
10	30.7650	15807.27	30.7544	15812.07	25.0330	18535.93
20	40.1712	18479.99	40.1607	18486.22	34.4393	22116.14
50	47.4896	20542.12	47.4791	20549.46	41.7577	24878.46
100	48.9933	20963.95	48.9827	20971.52	43.2613	25443.52
			$(a=3,\ b=2)$			
5	21.9605	13282.68	21.9500	13286.11	16.2286	15154.12
10	33.4356	16568.66	33.4250	16573.87	27.7036	19555.84
20	42.8418	19234.25	42.8313	19240.88	37.1099	23126.51
50	50.1602	21290.88	50.1497	21298.62	44.4283	25881.46
100	51.6639	21711.59	51.6533	21719.56	45.9319	26445.01

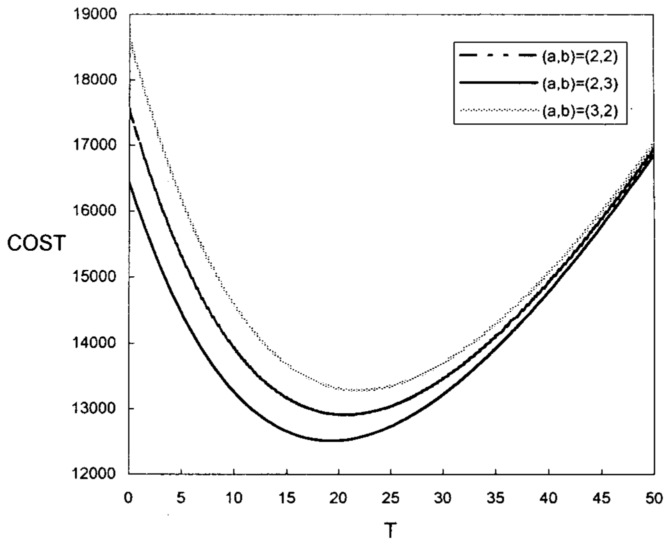

Figure 1. Total expected software cost $C_2(T)$ with $T_w = 5$, $C_t \sim TN(300,100^2,250,500)$ in Case 2.

Reference

[1] Goel, A. L. and Okumoto, K.(1979), Time-Dependent Error-Detection Rate Model for Software Reliability and Other Performance Measures, *IEEE Transactions on Reliability*, vol.28, pp.206-211.

[2] Juang, M. G. and Anderson, G.(2004), A Bayesian Method on Adaptive Preventive Maintenance Problem, *European Journal of Operational Research*, vol. 155, pp.455-473.

[3] Kimura, M., Toyota, T. and Yamada, S.(1999), Economic Analysis of Software Release Problems with Warranty Cost and Reliability Requirement, *Reliability Engineering & System Safety*, vol. 66, pp.49-55.

[4] Leung, Y. W.(1992), Optimum Software Release Time with a Given Cost Budget, *The Journal of Systems and Software*, vol. 17, pp.233-242.

[5] Mazzuchi, T. A. and Soyer, R.(1988), A Bayes Empirical-Bayes Model for Software Reliability, *IEEE Transactions on Reliability*, vol.37, pp.248-254.

[6] McDaid, K. and Wilson, S. P.(2001), Deciding How Long to Test Software, *Journal of The Royal Statistical Society, Series D*, vol. 50, pp.117-134.

[7] Morail, N. and Soyer, R.(2003), Optimal Stopping in Software Testing, *Naval Research Logistics*, vol.50, pp .88-104.

[8] Okumoto, K. and Goel, A. L.(1980), Optimum Release Time for Software Systems based on Reliability and Cost Criteria, *The Journal of Systems and Software*, vol.1, pp.315-318.

[9] Pham, H.(1996), A Software Cost Model with Imperfect Debugging, Random Life Cycle and Penalty Cost, *International Journal of Systems Science*, vol. 27, pp.455-463.

[10] Yamada, S. and Osaki, S.(1985), Cost–Reliability Optimal Release Policies for Software Systems, *IEEE Transactions on Reliability*, vol.34, pp.422-424.